

CLAIMS

1. (Previously Presented) A method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:
 - enabling receipt of information describing two or more types of protection;
 - enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear and wherein said portions of code are not required to be associated with one or more object oriented classes; and
 - enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.
2. (Cancelled)
3. (Original) The method of Claim 1, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.
4. (Original) The method of Claim 3, wherein said one or more domain attributes includes a domain identifier that specifies a unique value for a particular domain.
5. (Original) The method of Claim 3, wherein said one or more domain attributes includes a Private Key that specifies a unique value for protecting each user that concurrently uses a particular domain.

6. (Original) The method of Claim 3, wherein said one or more domain attributes includes a SharedCode Key that specifies a value that a particular domain must use to access code associated with another domain.
7. (Original) The method of Claim 3, wherein said one or more domain attributes includes a SharedData Key that specifies a value that a particular domain must use to access data associated with another domain.
8. (Original) The method of Claim 3, wherein said one or more domain attributes includes an AllowOthers that specifies a value that a particular domain must use to access code associated with another domain in conjunction with said particular domain performing cross-domain switching to said other domain.
9. (Original) The method of Claim 3, wherein said one or more domain attributes includes an AccessOthers Key that specifies a value that is used to request access of code associated with a particular domain on behalf of another domain.
10. (Previously Presented) A method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:
 - detecting a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level of said computer system and wherein said portions of code are not required to be associated with one or more object oriented classes;
 - determining whether said first portion of code is allowed to access said second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear; and
 - if said relationship specifies that said first portion of code may access said second portion of code, then

allowing said first portion of code to access said second portion of code;
else
not allowing said first portion of code to access said second portion of code.

11. (Original) The method of Claim 10, wherein said information describing said two or more types of protection and said information describing said relationships are associated with said portions of code and wherein the method further comprises retrieving said information describing said two or more types of protection and said information describing said relationships .

12. (Previously Presented) A computer system comprising:
a memory unit; and
a processor coupled to the memory unit, the processor for executing a method for enforcing protection in a computer system by decoupling protection from privilege, the method comprising:
enabling at a user interface receipt of information describing two or more types of protection;
enabling at the user interface receipt of information describing a relationship between said two or more types of protection and portions of code are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear and wherein said portions of code are not required to be associated with one or more object oriented classes; and
enabling at a link-editor the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.

13. (Original) The computer system of Claim 12, wherein said relationship is user definable.

14. (Original) The computer system of Claim 12, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.

15. (Previously Presented) A computer system comprising:
a memory unit; and
a processor coupled to the memory unit, the processor for executing a method for providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:
detecting at a memory manager a request from a first portion of code to access a second portion of code, wherein said first and second portions of code are executed in a same privilege level of said computer system and wherein said portions of code are not required to be associated with one or more object oriented classes;
determining at said memory manager whether said first portion of code is allowed to access said second portion of code based on information describing two or more types of protection and also based on information describing a relationship between said two or more types of protection and said portions of code, wherein said relationship is not required to be linear; and
if said relationship specifies that said first portion of code may access said second portion of code, then
allowing at said memory manager said first portion of code to access said second portion of code;
else
not allowing at said memory manager said first portion of code to access said second portion of code.

16. (Original) The computer system of Claim 15, wherein said information describing said two or more types of protection and said information describing

said relationships are associated with said portions of code and wherein the method further comprises retrieving at a loader said information describing said two or more types of protection and said information describing said relationships.

17. (Previously Presented) A computer-readable medium having computer-readable program code embodied therein for causing a computer system to perform a method of providing flexible protection in a computer system by decoupling protection from privilege, the method comprising:
 - enabling receipt of information describing two or more types of protection;
 - enabling receipt of information describing a relationship between said two or more types of protection and portions of code that are executed in a same privilege level of the computer system, wherein said relationship is not required to be linear and wherein said portions of code are not required to be associated with one or more object oriented classes; and
 - enabling the association of said information describing said two or more types of protection and said information describing said relationship with said portions of code, wherein a first portion of code allowing a second portion of code to access the first portion of code does not depend on the second portion of code allowing the first portion of code to access the second portion of code.
18. (Original) The computer-readable medium of Claim 17, wherein said relationship is user definable.
19. (Original) The computer-readable medium of Claim 17, wherein said portions of code are domains and each of said types of protection is defined at least in part by one or more domain attributes.
20. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes a domain identifier that specifies to a unique value for a particular domain.

21. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes a Private Key that specifies a unique value for protecting each user that concurrently uses a particular domain.
22. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes a SharedCode Key that specifies a value that a particular domain must use to access code associated with another domain.
23. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes a SharedData Key that specifies a value that a particular domain must use to access data associated with another domain.
24. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes an AllowOthers that specifies a value that a particular domain must use to access code associated with another domain in conjunction with said particular domain performing cross-domain switching to said other domain.
25. (Original) The computer-readable medium of Claim 19, wherein said one or more domain attributes includes an AccessOthers Key that specifies a value that is used to request access of code associated with a particular domain on behalf of another domain.
26. (Previously Presented) The computer system of Claim 15, wherein said second portion of code is allowed to access said first portion of code after a third portion of code accesses said second portion of code and wherein said third portion of code is not required to allow access to said first portion of code.